

Asynchronous sampled-data synchronization with small communications delays

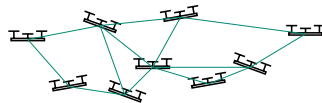
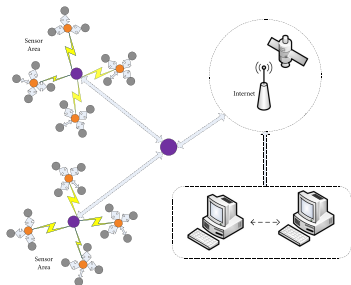
Gal Barkai Leonid Mirkin Daniel Zelazo

Technion - Israel Institute of technology

63rd IEEE Conference on Decision and Control

Networked multi-agent systems

- Multiple dynamic units interacting over a **network**
- Collective goals under limited **communication resources**
- Many applications
 - É sensor networks
 - É home automation
 - É multi-robot coordination



Archetype NMAS objective - State Agreement

Agents:

$$\dot{G}_g^1 \mathcal{L}^0 = G_g \quad D_g^1 \mathcal{L}^0 - \delta \geq \epsilon \quad \forall g \in \{1, \dots, n\}$$

with $\delta \in \mathbb{R}^+$ and $\epsilon \in \mathbb{R}^+$.

Goal: design distributed control signals $D_g^1 \mathcal{L}^0$ such that the states synchronize, i.e.

$$\lim_{t \rightarrow \infty} \|G_g^1 \mathcal{L}^0 - G_g^1 \mathcal{L}^0\| = 0 \quad \forall g \in \{1, \dots, n\}$$

with some trajectory generated by $A^1 \mathcal{L}^0 = \theta_0 A^1 \mathcal{L}^0$, for a given $\theta_0 \in \mathbb{R}^+$ with $\text{spec}(A^1 \mathcal{L}^0) \subset \mathbb{C}_0$, i.e.

$$\lim_{t \rightarrow \infty} \|G_g^1 \mathcal{L}^0 - A^1 \mathcal{L}^0\| = 0 \quad \forall g \in \{1, \dots, n\}$$

- θ_0 is an arbitrary model representing the target trajectory.

Archetype NMAS objective - State Agreement

Agents:

$$\mathcal{G}_g^1 \mathcal{L}^0 = G_g, \quad D_g^1 \mathcal{L}^0 = \mathcal{B} \gg 1 - \dots - \mathcal{A}^1$$

with $\mathcal{B} \in \mathbb{R}^{n \times n}$ and $\mathcal{A}^1 \in \mathbb{R}^{n \times n}$.

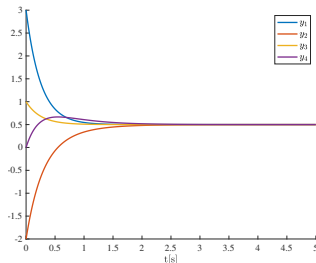
Goal: design distributed control signals $D_g^1 \mathcal{L}^0$ such that the states synchronize, i.e.

$$\lim_{t \rightarrow \infty} \|G_g^1 \mathcal{L}^0 - G_g^1 \mathcal{L}^0\| = 0 \quad \forall g \in \mathcal{G}$$

with some trajectory generated by $A^1 \mathcal{L}^0 = A_0^1 \mathcal{L}^0$, for a given $A_0^1 \in \mathbb{R}^{n \times n}$ with $\text{spec}(A_0^1) \subset \mathbb{C}_0$, i.e.

$$\lim_{t \rightarrow \infty} \|G_g^1 \mathcal{L}^0 - A^1 \mathcal{L}^0\| = 0 \quad \forall g \in \mathcal{G}$$

- A_0^1 is an arbitrary model representing the target trajectory.



For $A_0^1 = A_0^1 = 0$

$A^1 \mathcal{L}^0 = \text{const}$

Consensus problem

Archetype NMAS objective - State Agreement

Agents:

$$G_{\delta}^1 \mathcal{L}^0 = G_{\delta} \quad D_{\delta}^1 \mathcal{L}^0 = \delta \mathbb{1} - \dots - \mathcal{A}^1$$

with $\delta \in \mathbb{R}^+$ and $\mathbb{1} \in \mathbb{R}^+$.

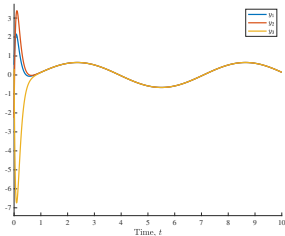
Goal: design distributed control signals $D_{\delta}^1 \mathcal{L}^0$ such that the states synchronize, i.e.

$$\lim_{\delta \rightarrow 0} \|G_{\delta}^1 \mathcal{L}^0 - G_{\delta}^1 \mathcal{L}^0\| = 0 \quad \delta \in \mathbb{R}^+$$

with some trajectory generated by $A^1 \mathcal{L}^0 = A_0^1 \mathcal{L}^0$, for a given $A_0^1 \in \mathbb{R}^{n \times n}$ with $\text{spec}(A_0^1) \subset \mathbb{C}_0$, i.e.

$$\lim_{\delta \rightarrow 0} \|G_{\delta}^1 \mathcal{L}^0 - A_0^1 \mathcal{L}^0\| = 0 \quad \delta \in \mathbb{R}^+$$

- A_0^1 is an arbitrary model representing the target trajectory.



For $A_0^1 =$
 $A^1 \mathcal{L}^0$ is time-varying
Synchronization

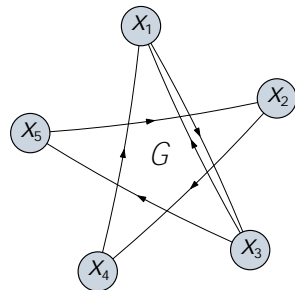
Communication constraints

The controller must respect three types of constraints:

Communication constraints

The controller must respect three types of constraints:

Spatial constraints: an agent can transmit only within its neighborhood $N_g^1 \mathcal{L}^0$. The neighborhoods can be **time-varying** and communication **directed**.

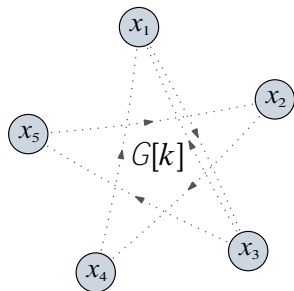


Communication constraints

The controller must respect three types of constraints:

Spatial constraints: an agent can transmit only within its neighborhood $N_g^1 \mathcal{L}^0$. The neighborhoods can be time-varying and communication directed.

Temporal constraints: agents communicate only at discrete sampling instances, $\ell \in \mathcal{B}; g$. The sampling can be **aperiodic** and **asynchronous**.



Communication constraints

The controller must respect three types of constraints:

Spatial constraints: an agent can transmit only within its neighborhood $N_g^1 \mathcal{C}^0$. The neighborhoods can be time-varying and communication directed.

Temporal constraints: agents communicate only at discrete sampling instances, $\mathcal{C} \in \mathcal{B}; g$. The sampling can be aperiodic and asynchronous.

Transmission delays: the information transmitted by agent g at $\mathcal{C} = \mathcal{B};$ arrives to agent g at $\mathcal{C} = \mathcal{C}_{gg} : \mathcal{N} := \mathcal{B}; \mathcal{C}_{gg} : \mathcal{N}$. The delays can be **time-varying** and **heterogeneous**.

These pose significant **challenges** for control design.

Basic assumptions

- A_1 : The pair (A, B) is stabilizable and there exists a gain K such that $A - BK$ is stable.

Solvability

The agents are stabilizable and the trajectory is attainable.

Basic assumptions

- A_1 : The pair $(\mathcal{A}, \mathcal{B})$ is stabilizable and there exists a gain K such that $\mathcal{A} - BK$ is stable.
- A_2 : there is a strictly increasing sub-sequence of sampling indices $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for all $k \in \mathbb{N}$
 - 1 the intervals $B_{k, f(k)} \dots B_{k, k}$ are uniformly bounded;
 - 2 $\mathcal{D}_{k, f(k)} \mathcal{G}_{k, f(k)}$ contains a directed rooted tree.

Joint connectivity

Information persistently propagates through the network

Basic assumptions

- A_1 : The pair (A, B) is stabilizable and there exists a gain K such that $A - BK$ is stable.
- A_2 : there is a strictly increasing sub-sequence of sampling indices $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for all $k \in \mathbb{N}$
 - 1 the intervals $B_{k, f(k)-1} \dots B_{k, k}$ are uniformly bounded;
 - 2 $\mathcal{D}_{k, f(k)-1}^{k, f(k)}$ contains a directed rooted tree.
- A_3 : incoming information is time stamped and

$$B_{k, f(k)-1} \dots B_{k, k} \in \mathbb{R}^{n \times m} \quad \forall k \in \mathbb{N} \quad f(k) \in \mathbb{N}.$$

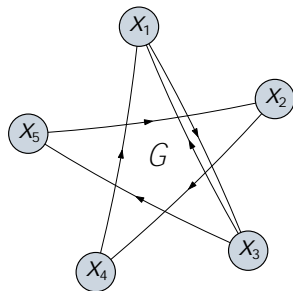
Small delays

The delays can be locally calculated and are *small* compared to the sampling interval.

The standard approach to agreement problems

Sequential design:

- 1 Transmit the state, $G_\theta^1 \mathcal{L}^0$, and use a consensus-based structure to satisfy the spatial constraints.

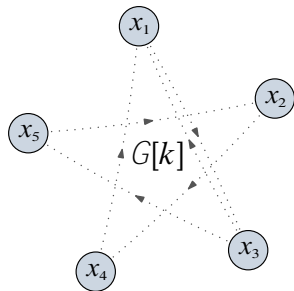


$$D_\theta^1 \mathcal{L}^0 = \begin{matrix} \tilde{O} & & \\ & {}^1 G_\theta^1 \mathcal{L}^0 & G_\theta^1 \mathcal{L}^{00} \\ & {}^2 N_\theta^1 \mathcal{L}^0 & \end{matrix}$$

The standard approach to agreement problems

Sequential design:

- 1 Transmit the state, $G_\theta^{-1} \mathcal{L}^0$, and use a consensus-based structure to satisfy the spatial constraints.
- 2 Assume the sampling is periodic and the control signal constant between samples.



$$D_\theta^{-1} \mathcal{L}^0 = \begin{matrix} \tilde{O} \\ \begin{matrix} G_\theta^{-1} B_\theta^0 & G_\theta^{-1} B_\theta^{00} \end{matrix} \\ 92 N_\theta \times \mathbb{K} \end{matrix}$$

The standard approach to agreement problems

Sequential design:

- 1 Transmit the state, $G^1 \mathcal{L}^0$, and use a consensus-based structure to satisfy the spatial constraints.
- 2 Assume the sampling is periodic and the control signal constant between samples.
- 3 Treat delays and sampling variations as perturbations.

$$D^1 \mathcal{L}^0 = \begin{matrix} \tilde{O} \\ \begin{matrix} G^1 B^0 & G^1 B^{00} \end{matrix} \\ 92 N_g : \mathbb{K} \end{matrix}$$

The standard approach to agreement problems

Sequential design:

- 1 Transmit the state, $G_g^1 \mathcal{L}^0$, and use a consensus-based structure to satisfy the spatial constraints.
- 2 Assume the sampling is periodic and the control signal constant between samples.
- 3 Treat delays and sampling variations as perturbations.
- 4 Solve a **robust control problem** to find an appropriate gain.

$$D_g^1 \mathcal{L}^0 = \begin{matrix} \tilde{O} \\ \tilde{G}_g^1 B :^0 & G_g^1 B :^{00} \\ 92 N_g \times \mathbb{K} \end{matrix}$$

The standard approach to agreement problems

Sequential design:

- 1 Transmit the state, $G_g^1 \mathcal{L}^0$, and use a consensus-based structure to satisfy the spatial constraints.
- 2 Assume the sampling is periodic and the control signal constant between samples.
- 3 Treat delays and sampling variations as perturbations.
- 4 Solve a robust control problem to find an appropriate gain.

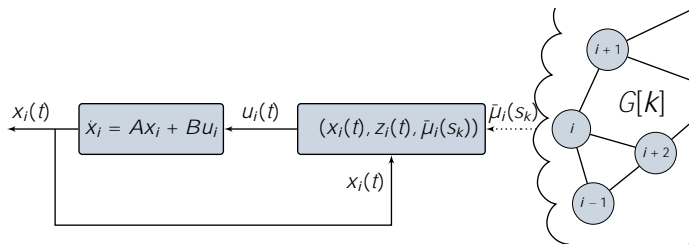
This often

- induces conservatism (discretization, input-delay)
- scales badly (# of decision variables may scale with a)
- does not exploit the spatio-temporal interplay of the problem.

$$D_g^1 \mathcal{L}^0 = \begin{matrix} \tilde{O} \\ \tilde{G}_g^1 B :^0 & G_g^1 B :^{00} \\ 92 N_g : \frac{1}{4} \end{matrix}$$

A key observation: spatio-temporal interplay

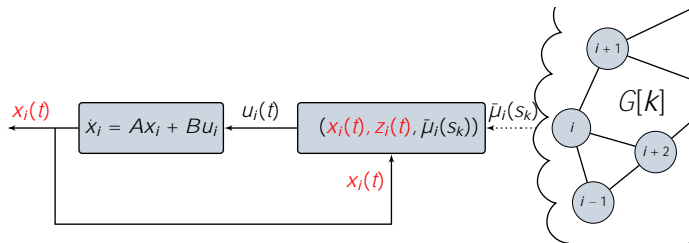
Inter-neighbor interactions and local actions occur at **different time scales**.



A key observation: spatio-temporal interplay

Inter-neighbor interactions and local actions occur at **different time scales**.

- **Local information:** the state $G_8^1 \mathcal{L}^0$ and controller variable $I_8^1 \mathcal{L}^0$, are continuously available.
 - Not effected by **communication constraints!**



A key observation: spatio-temporal interplay

Inter-neighbor interactions and local actions occur at **different time scales**.

- **Local information:** the state $G_g^1 \mathcal{L}^0$ and controller variable $I_g^1 \mathcal{L}^0$, are continuously available.
 - Not effected by **communication constraints!**
- **Neighboring information:** transmitted information, \mathcal{B}_g^1 , is available intermittently and may be delayed.

A key observation: spatio-temporal interplay

Inter-neighbor interactions and local actions occur at **different time scales**.

- **Local information:** the state $G^1 \mathcal{L}^0$ and controller variable $I^1 \mathcal{L}^0$, are continuously available.
 - Not effected by **communication constraints!**
- **Neighboring information:** transmitted information, \mathcal{B}^0 , is available intermittently and may be delayed.

A hybrid controller **not** based on a discretized consensus protocol can exploit the interplay between local and global information.

The paradigm: two measurements for two goals

- **Flow dynamics:** use local information to stabilize the system around a local *emulation* of the required trajectory $\hat{g}^1 \mathcal{L}^0$ via a **continuous** control signal.

The paradigm: two measurements for two goals

- **Flow dynamics:** use local information to stabilize the system around a local *emulation* of the required trajectory $\hat{g}^1 \mathcal{L}^0$ via a **continuous** control signal.

- **Jump dynamics:** at sampling instances, use incoming information, $\hat{g}^1 \mathcal{B} : \mathbb{R}^2 \rightarrow \mathbb{N}_g \gg \frac{1}{4}$, to update the emulators **discretely**.

The paradigm: two measurements for two goals

- **Flow dynamics:** use local information to stabilize the system around a local *emulation* of the required trajectory ℓ^0 via a **continuous** control signal.

$$\|G\ell^0 - \ell^0\| \ll \delta$$

- **Jump dynamics:** at sampling instances, use incoming information, $B^0 - 2N\delta \ll \epsilon$, to update the emulators **discretely**.

$$\|B^0 - B^0\| \ll \delta - \epsilon$$

- Two objectives: flow dynamics should **track**, and jump dynamics should **synchronize**.
É This was the idea behind (Barkai, Mirkin, and Zelazo, 2023).

The hybrid controller for the delay-free problem

Theorem (Barkai, Mirkin, and Zelazo, 2023)

If $A_{1,2}$ holds, A_d is Hurwitz, and $0 = \dots$; then local control law

$$D_\delta^1 \mathcal{L}^0 = A_d G_\delta^1 \mathcal{L}^0 + \frac{1}{a} (A_d^{0,1} I_\delta^1 \mathcal{L}^0 + G_\delta^1 \mathcal{L}^{0,0})$$

generated by a hybrid controller with the following *flow* and *jump* dynamics

$$\begin{aligned} \dot{x} &= A_\delta^1 \mathcal{L}^0 = A_d G_\delta^1 \mathcal{L}^0 + \frac{1}{a} (A_d^{0,1} I_\delta^1 \mathcal{L}^0 + G_\delta^1 \mathcal{L}^{0,0}) \\ I_\delta^1 B_\delta^0 &= I_\delta^1 B_\delta^0 + \frac{1}{a} (A_d^{0,1} I_\delta^1 B_\delta^0 + I_\delta^1 B_\delta^0 + G_\delta^1 B_\delta^0 + G_\delta^1 B_\delta^{0,0}) \end{aligned} \quad (1)$$

asymptotically synchronize the agents to the required trajectory.

- Here the emulator is a linear combination of the agent and controller states

$$x_\delta := \frac{1}{a} I_\delta^1 \mathcal{L}^0 + G_\delta^0 \mathcal{L}^0$$

Transmission delays in the hybrids setup

- The delays affect only transmitted information \Rightarrow they do not affect the **flow**.

Transmission delays in the hybrids setup

- The delays affect only transmitted information \Rightarrow they do not affect the flow.
- It can be shown that each emulator evolves according to

$$\begin{aligned} \dot{x}_g^1 C^0 &= 0 \quad \dot{x}_g^1 C^0 - \dot{x}_g^1 0^0 = \dot{x}_g^1 0^0 \\ \dot{x}_g^1 C_{gg} : \frac{1}{a} &= \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 B : \frac{1}{a} \end{aligned}$$

equivalently, the jump dynamics can be written as

$$\dot{x}_g^1 C_{gg} : \frac{1}{a} = e^{-a g_{gg} : \frac{1}{a}} \dot{x}_g^1 B : \frac{1}{a} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 B : \frac{1}{a} \quad e^{-a g_{gg} : \frac{1}{a}} \dot{x}_g^1 B : \frac{1}{a}$$

Transmission delays in the hybrids setup

- The delays affect only transmitted information \Rightarrow they do not affect the flow.
- It can be shown that each emulator evolves according to

$$\begin{aligned} \dot{x}_g^1 C^0 &= 0 \quad \dot{x}_g^1 C^0 - \dot{x}_g^1 0^0 = \dot{x}_g^1 0^0 \\ \dot{x}_g^1 C_{gg} : \frac{1}{a} &= \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 B : 0^0 \\ & \quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \end{aligned}$$

equivalently, the jump dynamics can be written as

$$\dot{x}_g^1 C_{gg} : \frac{1}{a} = e^{-\frac{1}{a} \tilde{O}} \dot{x}_g^1 B : 0^0 \quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 B : 0^0 \quad e^{-\frac{1}{a} \tilde{O}} \dot{x}_g^1 B : 0^0$$

Transmission delays in the hybrids setup

- The delays affect only transmitted information \Rightarrow they do not affect the flow.
- It can be shown that each emulator evolves according to

$$\begin{aligned} \dot{x}_g^1 C^0 &= 0 \quad \dot{x}_g^1 C^0 - \dot{x}_g^1 0^0 = \dot{x}_g^1 0^0 \\ \dot{x}_g^1 C_{gg} : \frac{1}{a} &= \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 B : 0^0 \\ &\quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 C_{gg} : \frac{1}{a} \quad \dot{x}_g^1 B : 0^0 \end{aligned}$$

equivalently, the jump dynamics can be written as

$$\dot{x}_g^1 C_{gg} : \frac{1}{a} = e^{-\frac{1}{a} \tilde{O}} \dot{x}_g^1 B : 0^0 \quad \frac{1}{a} \tilde{O} \quad \dot{x}_g^1 B : 0^0 \quad e^{-\frac{1}{a} \tilde{O}} \dot{x}_g^1 B : 0^0$$

We can exploit this to construct a simple predictor regardless of the gains a and β .

The main result

Theorem

If $A_{1,2,3}$ holds, A_d is Hurwitz, and $\tau_0 = \tau$; then the sampled-data controller (1) with jump map

$$I_{\mathbb{R}^n} \mathcal{C}_{gg} : \mathbb{R}^n \rightarrow \mathbb{R}^n = I_{\mathbb{R}^n} \mathcal{C}_{gg} : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad e^{-\tau A_d} \quad \tilde{O} \quad \begin{matrix} \mathbb{R}^n \\ \mathbb{R}^n \end{matrix} \quad \begin{matrix} \mathbb{R}^n \\ \mathbb{R}^n \end{matrix} \quad \begin{matrix} \mathbb{R}^n \\ \mathbb{R}^n \end{matrix}$$

asymptotically synchronize the agents for:

- all initial conditions;
- all sampling sequences,
- and all time-varying delays satisfying A_3 .

Moreover, the system synchronizes with the **same** trajectory as in the delay-free case.

- Recall that $\mathbb{R}^n \mathcal{C}^0 = \frac{1}{a} I_{\mathbb{R}^n} \mathcal{C}^0 \quad G_{\mathbb{R}^n} \mathcal{C}^0$.

Proof outline - the big picture

- 1 It can be shown that $G_{\delta^1} \mathcal{L}^0$ synchronize if and only if $\delta^1 \mathcal{L}^0$ synchronize, and that

$$\begin{aligned} \delta^1 \mathcal{L}^0 &= \delta^1 \mathcal{L}^0 - \delta^1 \mathcal{L}^0 \\ \delta^1 \mathcal{L}_{gg} : \mathbb{N}^0 &= \delta^1 \mathcal{L}_{gg} : \mathbb{N}^0 \quad \frac{1}{a} e^{0g_{gg} : \mathbb{N}} \tilde{O} \quad \delta^1 B : \mathbb{N}^0 \quad \delta^1 B : \mathbb{N}^0 \end{aligned}$$

Recall that $0 = \dots$

- 2 Equivalently, the new jump map can be written as

$$\delta^1 \mathcal{L}_{gg} : \mathbb{N}^0 = e^{0g_{gg} : \mathbb{N}} \delta^1 B : \mathbb{N}^0 \quad \frac{1}{a} \tilde{O} \quad \delta^1 B : \mathbb{N}^0 \quad \delta^1 B : \mathbb{N}^0$$

which looks almost exactly like the delay-free case.

Proof outline - the prediction

- 1 A_3 ensures that each agent can receive at most 1 update from each agent in each sampling interval.

Proof outline - the prediction

- 1 A_3 ensures that each agent can receive at most 1 update from each agent in each sampling interval.
- 2 Assume that there are $?$ delayed updates, and consider the sequence

$$B: \quad @_1 \gg : \frac{1}{4} \ddot{Y} \quad \ddot{Y} @_? \gg : \frac{1}{4} \ddot{Y} B: \dots_1$$

where $@_g \gg : \frac{1}{4}$ are the ordered version of $\mathcal{C}_{gg} \gg : \frac{1}{4}$.

Simulation's setup

- ① All Simulations involve $a = 3$ agent with the nominal sampling sequence randomized between 3 graphs and satisfy A_2 .

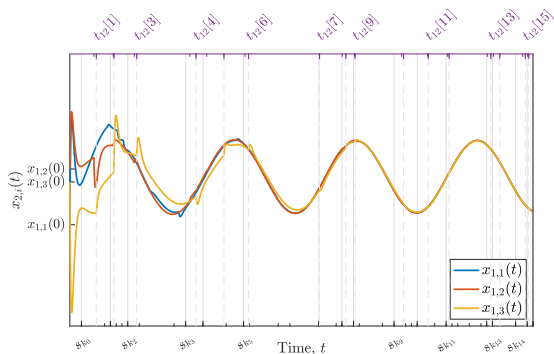
Simulation's setup

- 1 All Simulations involve $a = 3$ agent with the nominal sampling sequence randomized between 3 graphs and satisfy A_2 .
- 2 The nominal sampling intervals are randomized $B:_{s-1} \quad B: 2 \gg 0.3-1.8\%$, and the delays are generated by $g^1 B:_{s-1} \quad B: 0$, where g is uniformly distributed random variable from the interval $\gg 0-0.7\%$, hence satisfy A_3 .

Synchronization of LTI agents

Unstable LTI agents w/ time-varying trajectory

$$G_8^1 \mathcal{L}^0 = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix} \quad G_8^1 \mathcal{L}^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad D_8^1 \mathcal{L}^0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

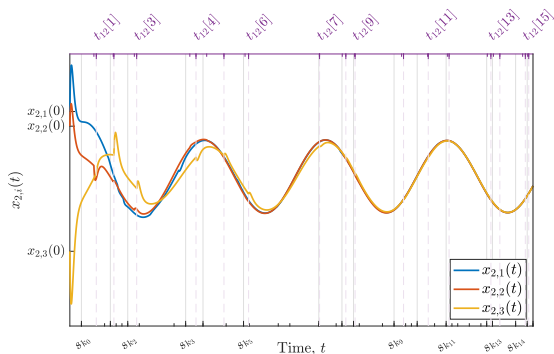


State's trajectories for first coordinate

Synchronization of LTI agents

Unstable LTI agents w/ time-varying trajectory

$$G_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix} \quad G_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad D_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

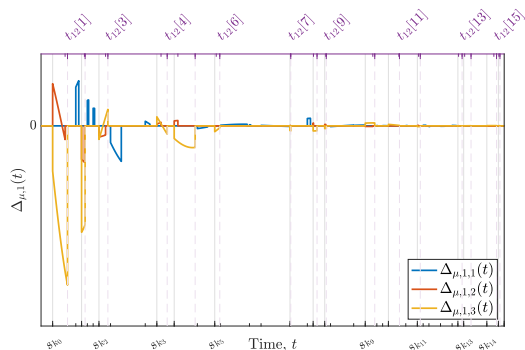


State's trajectories for second coordinate

Synchronization of LTI agents

Unstable LTI agents w/ time-varying trajectory

$$G_{\delta^1 \mathcal{L}^0} = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix}, \quad G_{\delta^1 \mathcal{L}^0}^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad D_{\delta^1 \mathcal{L}^0} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

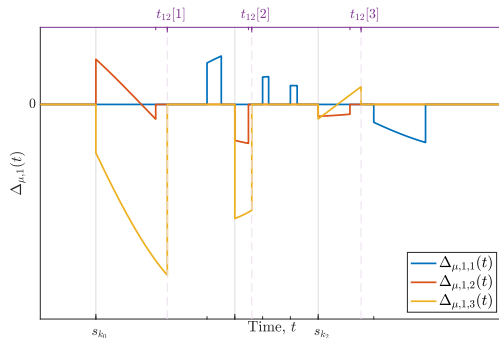


Error w/ delay-free system: $\delta_{-1}^1 \mathcal{L}^0 := \delta_{1}^1 \mathcal{L}^0 \delta_{1}^1 \mathcal{L}^0$

Synchronization of LTI agents

Unstable LTI agents w/ time-varying trajectory

$$G_{\delta^1 \mathcal{L}^0} = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix} \quad G_{\delta^1 \mathcal{L}^0}^{-1} = \begin{bmatrix} 2 & -2 \\ -1 & 1 \end{bmatrix} \quad D_{\delta^1 \mathcal{L}^0}^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

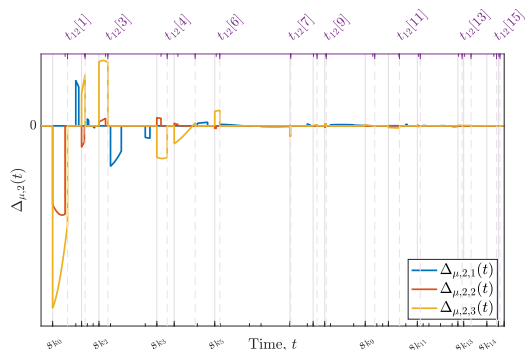


Error w/ delay-free system (zoomed): $\mathcal{L}^0 := \mathcal{L}^0 \quad \mathcal{L}^0$

Synchronization of LTI agents

Unstable LTI agents w/ time-varying trajectory

$$G_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 4 & 9 \\ 1 & 4 \end{bmatrix} \quad G_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad D_{\delta}^1 \mathcal{L}^0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



Error w/ delay-free system: $\mathcal{L}^0 := \mathcal{L}^0 \quad \mathcal{L}^0$

Concluding remarks

- Exploits the hybrid structure of the controller to compensate for transmission delays.
- Naturally works with heterogeneous and time-varying delays.
- The controller recovers the delay-free behavior - no loss of performance.
- It can be shown that each agent needs only a buffer of size 1 to implement the predictor.
- Future research:
 - É Robustness to uncertainty in $\mathcal{G}_{\theta g} : \frac{1}{4}$
 - É Extensions to longer time-delays
 - É Moving from state to output feedback.